

SOA: A Quality Attribute Perspective

Grace A. Lewis
glewis@sei.cmu.edu

SEI Webinar Series
June 23, 2011



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 23 JUN 2011		2. REPORT TYPE		3. DATES COVERED 00-00-2011 to 00-00-2011	
4. TITLE AND SUBTITLE SOA: A Quality Attribute Perspective				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 45	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

SEI Webinar Series

Keeping you informed of the latest solutions.



Software Engineering Institute

CarnegieMellon



Join the SEI Webinar Series on LinkedIn

This forum allows attendees to discuss or post questions from the presentations in the SEI Webinar Series, and submit suggestions for future topics.



Search for Group:
SEI Webinar

▶ www.linkedin.com



Software Engineering Institute

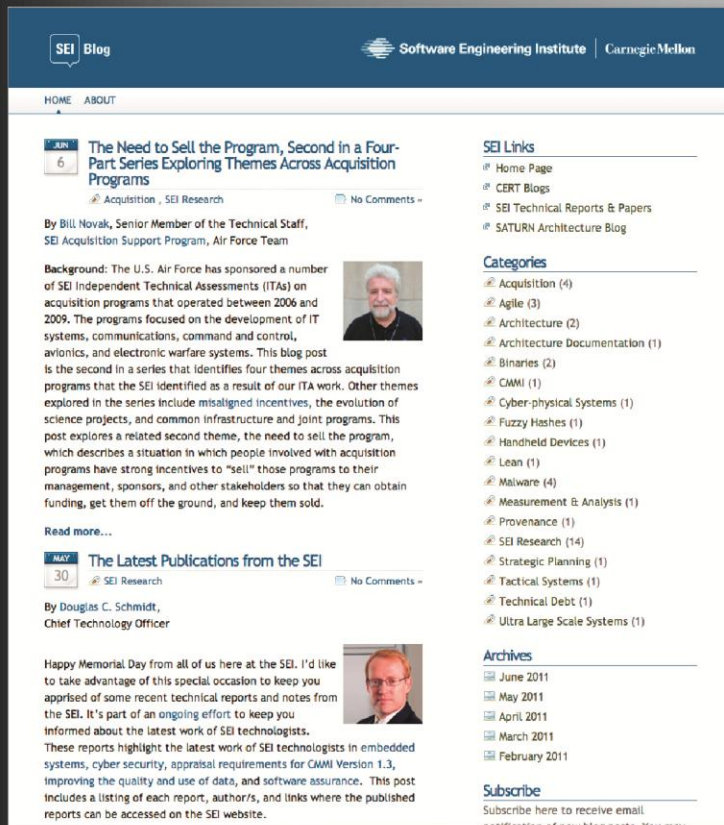
CarnegieMellon

June 2011 **Twitter #seiwebinar**
© 2011 Carnegie Mellon University



SEI Technology Blog

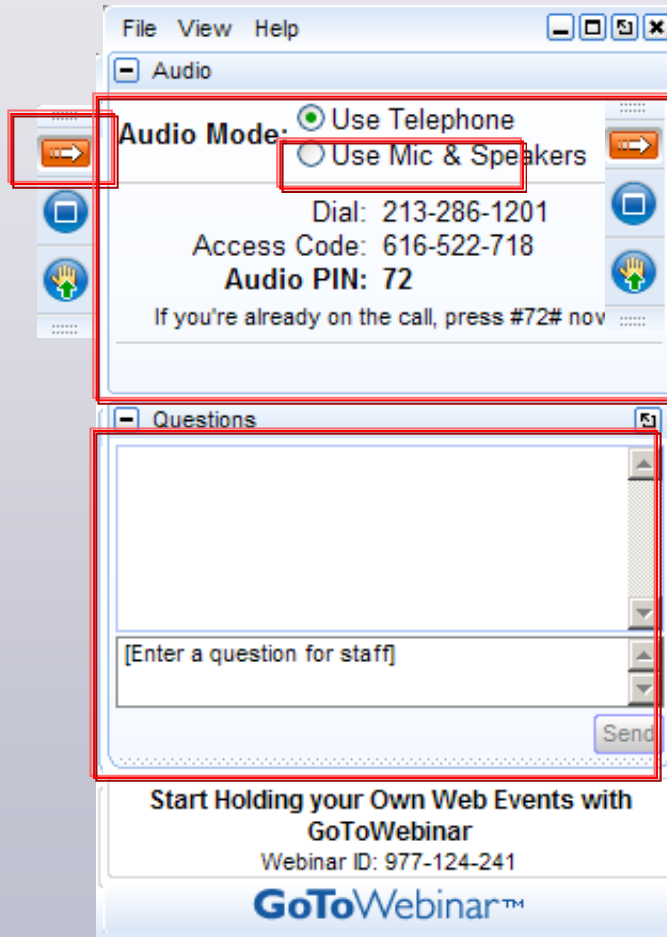
With posts written by staff members, the blog provides the SEI audience with insights into the broad spectrum of work at the SEI via a two-way, read-write medium.



<http://blog.sei.cmu.edu/>



How to Participate Today



Open and close your Panel

View, Select, and Test your audio

Submit text questions

Q&A addressed at the end of today's session



Today's Presenter

Grace Lewis is a senior member of the SEI technical staff within the Systems of Systems Practice (SoSP) Initiative in the Research, Technology, and Systems Solutions (RTSS) Program. Her current interests and projects are in service-oriented architecture (SOA), cloud computing, context-aware applications, and technologies for systems interoperability. Her latest publications include multiple reports and articles on these subjects and a book in the SEI Software Engineering Series to be published later in 2011. She is also a member of the technical faculty for the Master in Software Engineering Program at Carnegie Mellon University (CMU). Lewis holds a bachelor's degree in systems engineering and an Executive MBA from Icesi University in Cali, Colombia; and a master's degree in software engineering from CMU.



Agenda

Service-Oriented Architecture and Software Architecture: Review

Service-Orientation and Quality Attributes

Summary and Future Challenges



Polling Question 1

Which is the quality attribute that is most positively affected by the use of SOA concepts and technologies?

- Availability
- Modifiability
- Interoperability
- Performance
- Security



Polling Question 2

Which is the quality attribute that is most negatively affected by the use of SOA concepts and technologies?

- Modifiability
- Performance
- Reliability
- Scalability
- Security



Agenda

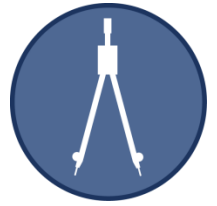
Service-Oriented Architecture and Software Architecture: Review

Service-Orientation and Quality Attributes

Summary and Future Challenges



What is SOA?



Service-oriented architecture is a way of designing, developing, deploying and managing systems, in which

- Services provide reusable business functionality via well-defined interfaces.
- Service consumers are built using functionality from available services.
- There is a clear separation between service interface and service implementation.
 - Service interface is just as important as service implementation.
- An SOA infrastructure enables discovery, composition, and invocation of services.
- Protocols are predominantly, but not exclusively, message-based document exchanges.



Services

Services are reusable components that represent business/operational tasks.

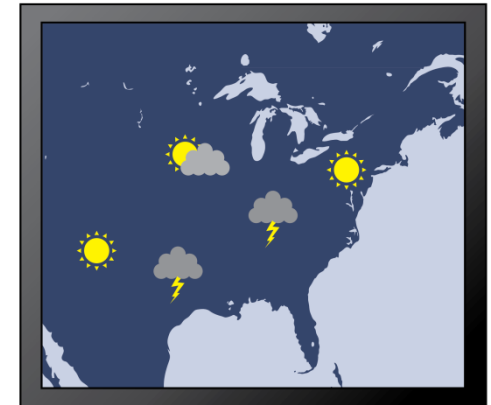
- Customer lookup
- Credit card validation
- Weather
- Hotel reservation



Services can be

- Globally distributed across organizations
- Reconfigured into new business processes

Service interface definitions are well-defined artifacts available in some form of service registry.



SOA Infrastructure

Set of technologies that bind service consumers to services

- Products, standards and protocols that support communication—typically message-based document exchanges
 - Web Services (WS*: HTTP, SOAP, WSDL; REST)
 - Message-oriented middleware (i.e. IBM Websphere MQ)
 - Publish/subscribe (i.e. Java Messaging Service — JMS)
 - CORBA ...
- Infrastructure services available to service providers and/or service consumers to perform common tasks or satisfy QoS requirements of the environment
 - Security, discovery, data transformation, ...



Service Consumers



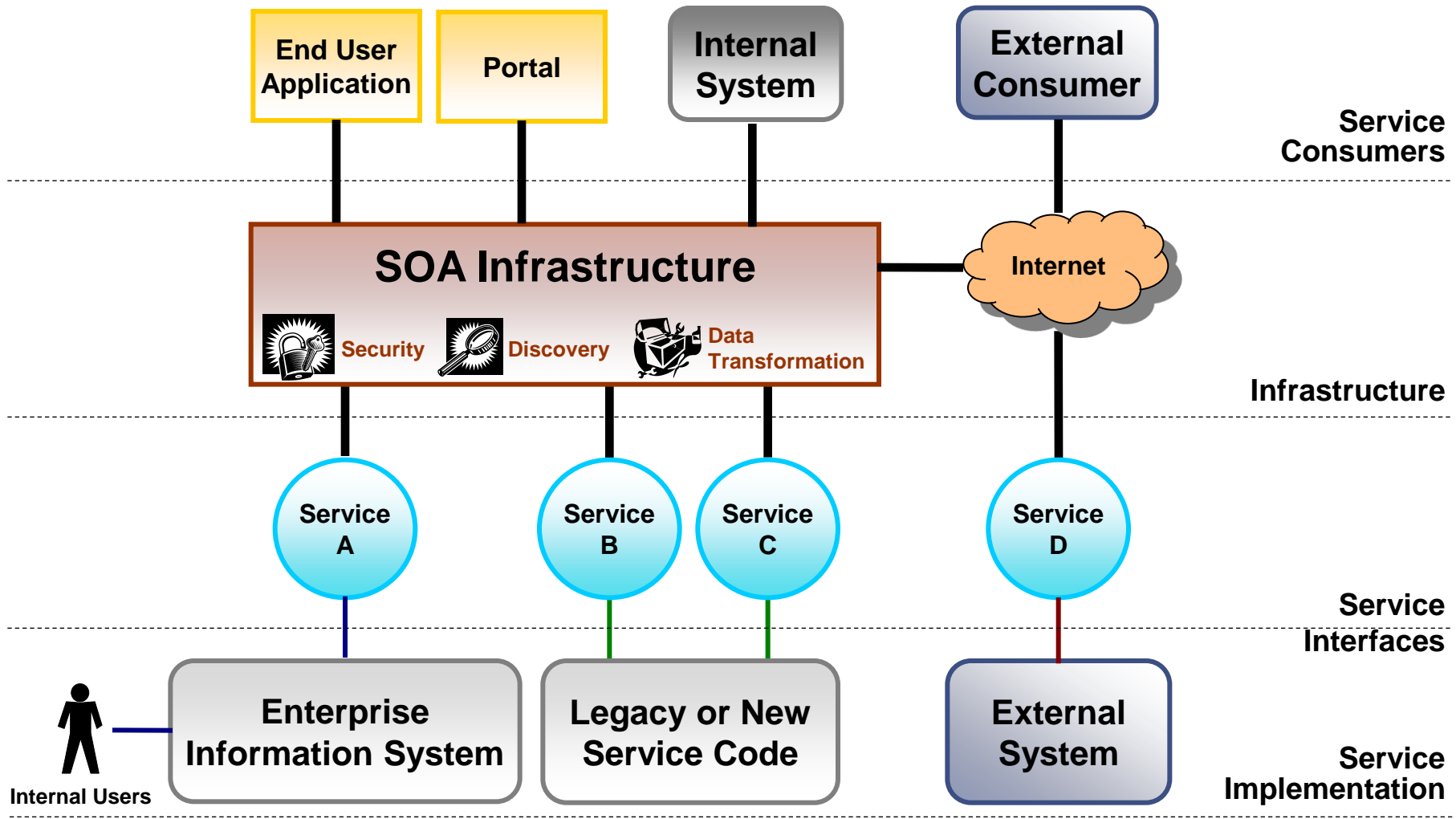
Clients for the functionality provided by the services

- End-user applications
- Internal systems
- External systems
- Composite services

Consumers programmatically bind to services.



Components of a Service-Oriented System



Benefits Associated with SOA Adoption



Cost-Efficiency

- Services provide functionality that can be reused many times by many consumers
- Services become a single point of maintenance and management for common functionality

Agility

- Via service discovery mechanisms, developers can find and take advantage of existing services to reduce development times

Legacy Leverage

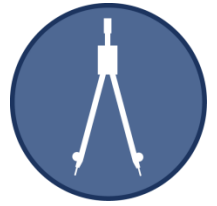
- Separation of service interface from service implementation provides true platform independence

Adaptability

- Separation of service interface from service implementation allows for incremental deployment of services and incremental modernization



Software Architecture



The current literature on software architecture offers many definitions

The definition we use is

A software architecture for a system is the structure or structures of the system, which comprise elements, their externally visible properties, and the relationships among them*

Why create a software architecture?

- Because a system's quality attributes can be predicted by studying its architecture

What is a quality attribute?

- A property of a system by which its quality will be judged by some stakeholder or stakeholders

* Bass, L.; Clements, P.; & Kazman, R. Software Architecture in Practice, Second Edition. Boston, MA: Addison-Wesley, 2003.



Sample Quality Attributes

Accessibility	Dependability	Nomadcity	Serviceability
Accountability	Deployability	Operability	Simplicity
Adaptability	Distributability	Performance	Stability
Administrability	Durability	Portability	Survivability
Affordability	Evolvability	Predictability	Sustainability
Agility	Extensibility	Recoverability	Tailorability
Auditability	Flexibility	Relevance	Testability
Availability	Installability	Reliability	Timeliness
Credibility	Interchangeability	Repeatability	Understandability
Compliant	Interoperability	Reproducibility	Usability
Composability	Learnability	Reusability	
Configurability	Maintainability	Safety	
Customizability	Manageability	Scalability	
Degradability	Mobility	Seamlessness	
Demonstrability	Modularity	Security	

What does
each one of
these mean?



Quality Attribute Scenarios



A fully-specified quality attribute scenario consists of six parts

- **Stimulus:** condition effecting the system
- **Response:** activity as a result of the stimulus
- **Source of Stimulus:** entity that generated the stimulus
- **Environment:** condition under which the stimulus occurred
- **Artifact stimulated:** artifact that was stimulated
- **Response measure:** measure by which the system's response will be evaluated

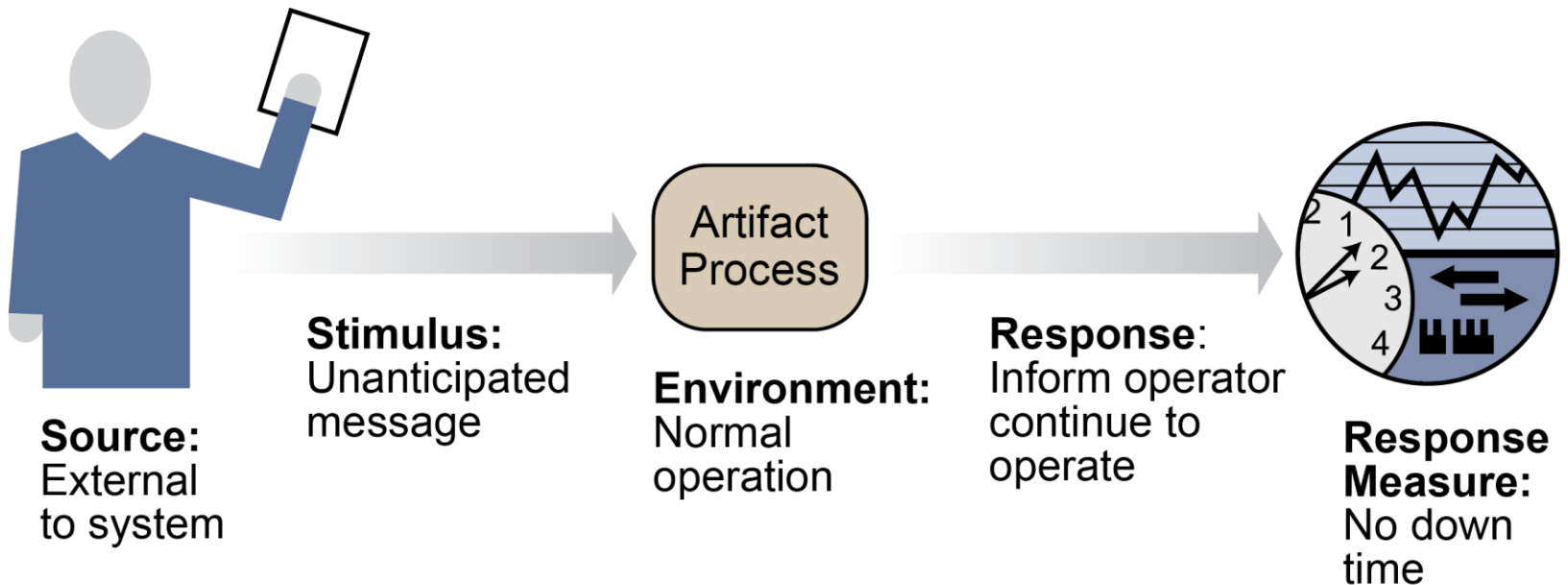
Quality attribute scenarios should be as specific as possible

- A good scenario makes very clear what the stimulus is and what the desired responses of the system are, in order to avoid ambiguity



Example Availability Scenario

An unanticipated external message is received by a process during normal operation. The process informs the operator of the receipt of the message and the system continues to operate with no down time.



Misconception: SOA Provides the Complete Architecture for a System

SOA is an architectural pattern/style/paradigm and not the architecture of the system itself.

- An architectural pattern provides guidance that embodies best practices.
 - The concrete elements and their interactions are the architecture of the system.
- Any number of systems can be developed based on an architectural pattern.
 - An architecture based on SOA inherits both the good and the bad.

Corollary: SOA cannot be bought off-the-shelf.

- System qualities have to be built into the architecture of the system.
- Decisions have to be made—service design and implementation, technologies, tradeoffs.



Agenda

Service-Oriented Architecture and Software Architecture: Review

Service-Orientation and Quality Attributes ←

Summary and Future Challenges

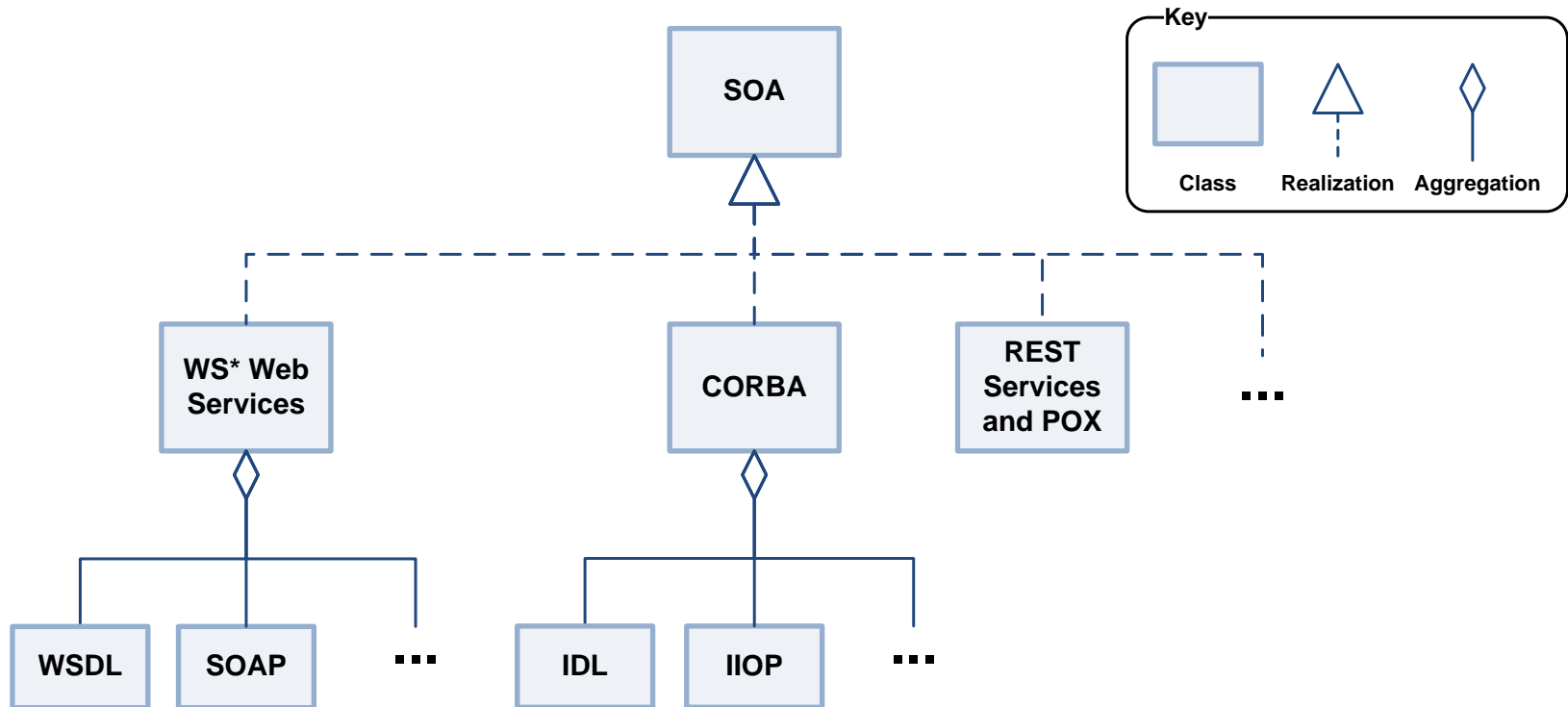


SOA is NOT a Specific Technology

SOA is an architectural pattern

- Systems that implement the SOA architectural pattern are called service-oriented systems

Web Services is **one** technology for SOA implementation



Quality Attributes in Service-Oriented Systems

The following slides are examples of common quality attribute scenarios for service-oriented systems, plus an analysis of how the SOA architectural pattern affects those qualities

The legend to indicate the effect is



The quality is positively affected



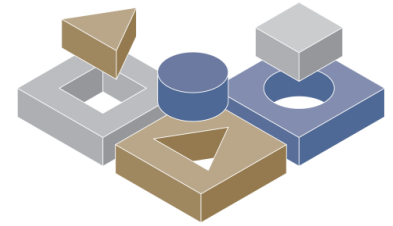
The quality is a challenge



It depends ...



Sample Interoperability Scenarios



Scenario I1

A new business partner that uses platform 'X' is able to implement a service consumer module that works with our available services in platform 'Y' in two person-days.

Scenario I2

A transaction in a legacy system running on platform 'X' is made available as a Web service to an enterprise application that is being developed for platform 'Y' using Web services technology. The wrapping of the legacy transaction as a service with proper security verification, transaction management, and exception handling is done in 10 person-days.



Interoperability



Interoperability refers to the ability of a collection of communicating entities to share specific information and operate on it according to agreed-upon operational semantics*

- Improved interoperability is a prominent benefit of service-orientation, especially when web services technology is considered
- Consumers and providers can easily use and provide services implemented in disparate platforms and different languages

Web Services promote cross-vendor and cross-platform interoperability

- Mature when basic standards are used: WSDL and SOAP
- Falls short when new standards are in the picture (e.g., WS-Security, WS-Transaction)

WS-I organization was created to promote interoperability (ws-i.org)

- The goal of WS-I profiles is to provide clarifications, refinements, interpretations and amplifications in areas of specific standards that are subject to multiple interpretations

* Brownsword et al. Current Perspectives on Interoperability. Software Engineering Institute. 2004.
<http://www.sei.cmu.edu/reports/04tr009.pdf>



Sample Performance Scenarios



Scenario P1

The service provider can process up to 'X' simultaneous requests during normal operation, keeping the response time on the server less than 'Y' seconds.

Scenario P2

The roundtrip time for a request from a service consumer in the local network to service 'X' during normal operation is less than 'Y' seconds.



Performance



Performance in an SOA context usually relates to response time or throughput and in most cases is negatively affected

- SOA is a distributed computing paradigm; the network increases response time
- Cross-platform interoperability requires intermediaries to do data marshalling and handle communication

In Web Services, use of XML impacts performance

- Studies show that XML messages can be 10 to 20 times larger than binary messages
- XML requires three CPU- and memory-intensive activities
 - Parsing
 - Validation
 - Transformation



Sample Availability Scenarios



Scenario A1

An improperly formatted message is received by a system during normal operation. The system records the message and continues to operate normally without any downtime.

Scenario A2

Unscheduled server maintenance is required on server 'X.'
The system remains operational in degraded mode for the duration of the maintenance.



Sample Scalability Scenarios



Scenario S1

Marketing landed several new high-volume accounts that will increase service request volume by a factor of 10. During normal operation, the service requests are processed without affecting the current quality of service.

Scenario S2

Marketing landed several new high-volume accounts that will increase service request volume by a factor of 10. During normal operation, the service requests are processed according to the Service-Level Agreements negotiated with each account.



Availability and Scalability



SOA solutions usually rely on mechanisms provided by the execution platform

- Horizontal scaling: Add load-balanced computer servers
- Vertical scaling: Increase capacity of a computer server
- Service scope: Configure when a new instance of a service should be created
 - Once to serve all requests (best if service implementation is reentrant or thread-safe)
 - For each new service consumer
 - For each new request

Another mechanism is to design services to be stateless

- To enable replication

SLAs for external services may define availability requirements



Sample Security Scenarios



Scenario 1

An attack is launched attempting to access confidential customer data. The attacker is not able to break the encryption used in all the hops of the communication and where the data is persisted. The system logs the event and notifies the system administrators.

Scenario 2

A request needs to be sent to a third-party service provider, but the provider's identity cannot be validated. The system does not make the service request and logs all relevant information. The third party is notified as well as with the system administrator.



Security



Some aspects of SOA solutions impact security

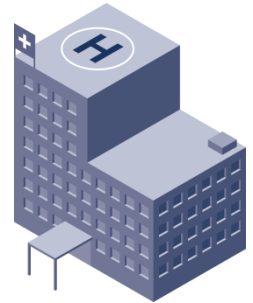
- Messages may be transmitted in text format and contain sensitive metadata
- If external services are used
 - Identity of service providers should be authenticated
 - Data should be transmitted and stored securely
- Data in the service registry should be reliable
- Authorization of clients to access services has to be configured

Transport security is usually addressed at the network level (e.g., SSL)

There are tradeoffs with interoperability, modifiability and performance



Sample Reliability Scenarios



Scenario R1

A sudden failure occurs in the runtime environment of a service provider. After recovery, all transactions are completed or rolled back as appropriate, such that the system maintains uncorrupted, persistent data.

Scenario R2

A service becomes unavailable during normal operation. The system detects the problem and restores the service within two minutes.



Reliability



Message reliability

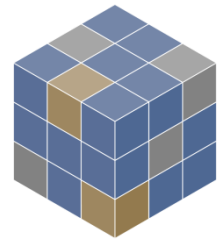
- Messages should be delivered in order and exactly once
- This is usually a concern of the SOA execution platform, not the service developer
- Available standards: WS-Reliability and WS-ReliableMessaging

Service reliability

- Goal is for the service to operate without failure or to report any failure
- Main issue is transaction management
 - Complex because of the distributed, multi-platform — and potentially multi-organizational — context
 - May require compensating transactions



Sample Modifiability Scenarios



Scenario M1

A service provider changes the service implementation, but the syntax and the semantics of the interface do not change. This change does not affect the service consumers.

Scenario M2

A service provider changes the interface of a service that is publicly available. The old version of the service is maintained for 12 months, and existing service consumers are not affected within that period.



Modifiability



Modifiability is the ability to make changes to a system quickly and cost effectively

- SOA promotes loose coupling between service consumers and providers
 - Services are modular and self-contained
 - There are few well-known dependencies, therefore reducing the cost of modifying the services
- Changing published interfaces can be a challenge, but SOA solutions deal with these changes through
 - Versioning mechanisms
 - Flexible contracts specified in XML
 - Special components in the infrastructure, such as the registry



Summary

Quality attributes have the strongest influence on architectural design decisions

- Quality attributes requirements can be captured as scenarios

SOA is a design pattern that promotes interoperability and modifiability, usually at the expense of performance and security

- SOA is not a complete architecture
- It is often combined with other patterns and tactics, e.g.
 - Brokers for data format and protocol translation
 - Routers for message routing based on runtime factors
 - Asynchronous queues for increased decoupling between consumers and providers
 - Registries (metadata centralization) for service discovery



Agenda

Service-Oriented Architecture and Software Architecture: Review

Service-Orientation and Quality Attributes

Summary and Future Challenges ←



SOA Adoption in Practice



What practice shows is that SOA is currently the best option available for systems integration and leverage of legacy systems

SOA has “crossed the chasm,” according to a recent Software AG user survey in which 90 percent of the respondents claim to have made some commitment to SOA adoption

- Have moved from early adopter to early majority

There is active research and development in many aspects related to SOA

- Probably too many aspects, which is also a problem

The bottom line is that organizations are still investing in SOA



However, We Want More!



As SOA is adopted within organizations and becomes a mainstream paradigm for systems development, requirements and expectations increase

The loosely-coupled, stateless, standards-based nature of the relationship between service consumers and service providers is changing to meet these new requirements

Global enterprises and the emerging market of third-party services and cloud computing are also placing expectations on service-oriented system architecture and design

- As organizations expand their systems to cross organizational boundaries, the requirements on their systems also expand

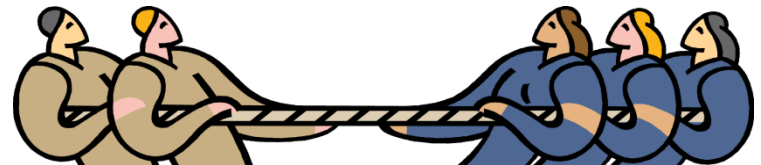


SOA Is Potentially Being Stretched Beyond Its Limits

What was initially an approach for asynchronous document-based message exchanges now has performance, availability, reliability, security, and other expectations of traditional distributed systems

The architect of service-oriented systems is going to play a crucial role

- Determining what expectations can or cannot be met by current SOA technologies
- Determining where tradeoffs can be made for the accomplishment of system qualities without having to sacrifice the loosely coupled, stateless, standards-based characteristics that have made SOA a worthwhile technology
- Performing early, contextual technology evaluation and continuous technology scouting



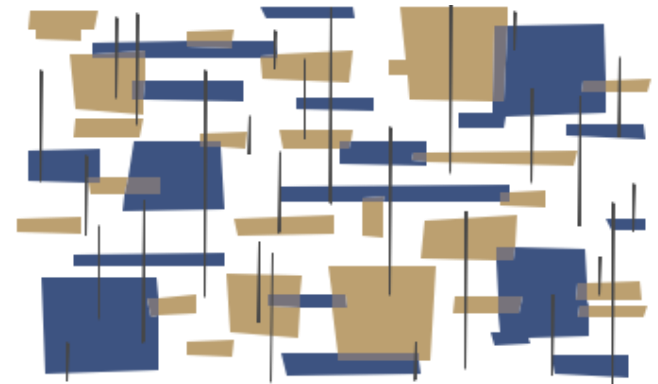
Need to Separate Service-Orientation from SOA Implementation Technologies

The concept of service-orientation is here to stay, but the technologies will change over time to meet new requirements

A challenge for architects of service-oriented systems is to reduce the impact of changing SOA technologies from the implementation of service-oriented concepts

- “Separation of concerns on steroids”

SOA is not dead!



Contact Information



Grace A. Lewis

Research, Technology and Systems Solutions (RTSS) Program
System of Systems Practice (SoSP) Initiative

Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Phone: +1 412-268-5851

Email: glewis@sei.cmu.edu

WWW: <http://www.sei.cmu.edu/about/people/glewis.cfm>



SOA Architect Professional Certificate



Course
Path



What Attendees Say
About SOA Courses



Meet Our
Instructors



Get Answers
Ask a Question



Explore SOA
Reports, Research...

Learn How to Architect Service-Oriented Systems

From the Experts at the Software Engineering Institute

Architect with confidence.

The SEI prepares SOA architects to make their own decisions about the benefits, potential risks, and pitfalls of SOA adoption.

All courses are technology agnostic.

They apply to service-oriented systems in general and do not favor any specific implementation technologies, tools, or products.

Ready-to-implement methods.

All courses in the sequence are oriented toward practitioners and designed to provide skills that can be applied on the job right away.



Two SOA instructors discuss three takeaways from the SOA Certificate.

<http://www.sei.cmu.edu/go/soa-certificate/>



Software Engineering Institute

Carnegie Mellon

June 2011 [Twitter #seiwebinar](#)
© 2011 Carnegie Mellon University

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

